IN THE CLAIMS:

Claim 3 has now been amended to include all of the limitations of claims 1 and 2.

Claims 4 and 5 are now dependent on the newly-amended claim 3.

The allowed claim 6 is also attached herein.

Please cancel claims 1, 2 and 7.

1. (Cancelled).

2. (Cancelled).

3. (Currently Amended) In a Thin Client Sizing Tool, a method for developing a Metafarm having an optimal number of Server Farms to provide recommended configurations meeting certain specified parameters, wherein a number of factors are established which include: (i) the total number of users who will be using the Metafarm; (ii) an Availability goal which indicates the percentage of time that the systems and applications in each Server Farm will be accessible to all the users involved; (iii) assigning a user weight volume to each type of user to indicate estimated average usage or light, medium, heavy or super heavy; (iv) calculating the number of servers to be assigned to each Server Farm and which will fulfill the said Availability goal; (v) calculating the number of redundant servers per Server Farm needed to provide maximum performance over and above the average nominal performance while still fulfilling said Availability goal; (vi) seeking to find the minimum number of Server Farms which still provide an optimum Redundancy Factor of extra servers which will still fulfill the desired Availability goal, comprising the steps of:

(a) delivering input data on the total number of users to be serviced, the Availability goal to be achieved, the User-Weight utilization factors involved, and the preferred Server types to be used;

(b) sequencing a series of calculations to determine the number of Servers per Farm and the number of redundant Servers per Farm which match or exceed the said Availability goal;

and wherein data on Benchmark operational parameters are consulted on a specific type of server to establish the maximum number of users which can be supported by said chosen specific type of server, and wherein step (b) includes the steps of:

(b1) retrieving a Benchmark parameter which indicates the maximum number of users which can be serviced by a chosen Server type;

(b2) calculating a preliminary number of such chosen Servers which will constitute a Server Farm; and

~~3. (Currently Amended) The method of claim~~ 2 wherein a desired Redundancy Factor is used to add enough extra servers, designated as a number of redundant servers per farm, to enable maximal user usage over nominal user usage and wherein an estimated Availability Level is set for each chosen Server Farm, and wherein step (b2) includes the steps of:

(b2a) [[(b3)]] calculating the number of redundant Servers per Farm according to a preliminary set percentage parameter for the Redundancy Factor;

(b2b) [[(b4)]] calculating the estimated Availability Level for the Server Farm chosen[[.]];

(c) displaying a set of recommendations which show the minimum number of Server Farms which have the optimum redundancy factor and meet the values needed for the Availability goal.

4. (Currently Amended) The method of claim 3 wherein a desired Availability Level goal of a certain maximum downtime value is checked to see if it meets said Availability Level goal according to the number of Server Farms first estimated and the number of servers-per-farm first estimated, and the number of redundant servers-per-farm first estimated, and wherein step (b) [[which]] includes the steps of:

> (b3) [[(b5)]] if step (b2b) [[(b4)]] Availability Level does not meet or exceed the Availability Level goal, then initiate a sequential loop by either incrementing or decrementing the number of Server Farms to re-calculate the number of Servers per Farm and number of redundant Servers per Farm which meet or exceed the Availability Level goal.

5. (Currently Amended). The method of claim 4 [[3]] wherein there is accommodated a Redundancy Factor indicating the possible number of extra users which can be accommodated in a Server Farm which is added to the nominal number of users-per-server without exceeding the maximum allowable users-per-server set by benchmarking data,

and wherein correlation is made between the number of users-per-Server Farm, the number of servers and the estimated Availability Level for each Server Farm and wherein there is established the total number of servers in the entire Metafarm of server farms, such that step (b) ~~and which~~ includes the steps of:

(b4) [[(b5)]] decrementing the Redundancy Factor until no acceptable recommendations are available;

(b5) [[(b6)]] incrementing the Redundancy Factor in steps of 1% to find the optimum Redundancy Factor;

(b6) [[(b7)]] storing configuration recommendations in an array indicating output displays of the number of Servers correlated to the number of Users per Farm with the estimated Availability Level, estimated yearly downtime, number of redundant Servers in the Metafarm and the total number of Servers in the Metafarm.

6. (Previously presented). In a Thin Client Sizing tool, a method for optimizing the number of Server Farms to provide the most efficient recommended configurations which provide a desired Availability Level goal and Redundancy Factor, wherein data is accumulated as to the number of Users to be involved, the Availability goal of maximum downtime permitted, and the usage weight load by each type of user; the number of servers to be utilized in each Farm, plus the number of redundant servers to be placed in each Server Farm to allow performance service beyond the nominal usage; utilizing an experienced benchmark value for types of servers involved to ensure that the number of servers-per-farm does not exceed the appropriate benchmark value for the type of server involved; providing a calculated Availability Level for the maximum allowable downtime which meets the downtime goal for Availability Level; reporting out a number of output recommendations which correlates sets of parameters which link the number of farms with the number of users-per-farm with the Estimated Availability, with the Estimated Downtime, with the number of Redundant Servers, and with the Total Number of Servers in order to select that set of parametric criteria which best fulfill a selected customer's requirements, comprising the steps of:

(a) inputting of data to indicate the number of users involved, the Availability goals, the user-weight factors, and preferred server types;

(b) calculating the number of Servers per Farm to be utilized;

(c) calculating the number of redundant Servers to be placed in each Server Farm;

(d) using a benchmark to check if the number of Servers per Farm from steps (b) and (c) exceed the benchmark values for the Servers involved;

(e) if step (d) indicates that the number of Servers per Farm does not exceed the benchmark value, then calculating the estimated Availability Level of the Server Farm;

(f) checking to see that the said calculated Availability Level meets or exceeds the Availability Level goal;

(g) if the Availability Level goal is not met or exceeded, then incrementing the number of Server Farms by "1";

(h) checking to see if the number of Server Farms is greater than 100 or not greater than 100;

(i) if the number of Server Farms is less than 100, then requesting through steps (b), (c), (d), (e), (f), (g), and (h) until step (h) indicates that the number of Server Farms is greater than 100;

(j) checking to see that the number of output recommendations is greater than "0";

(k) decrementing the Redundancy Factor in steps of 5% until no acceptable recommendations are available;

(l) incrementing the Redundancy Factor in steps of 1% to develop a set of recommendations which minimize the number of Server Farms while still supporting the number of users required and still meeting the Availability Level goal.

7. (Cancelled).